



Content Manager Version 8 Overview

by Cataldo Mega
Ken Nelson

IBM Content Manager
for
Universities and Libraries

October 26 - 27, 2004

IBM Forum Stuttgart, Pascalstrasse 100 70569 Stuttgart, Germany

Agenda

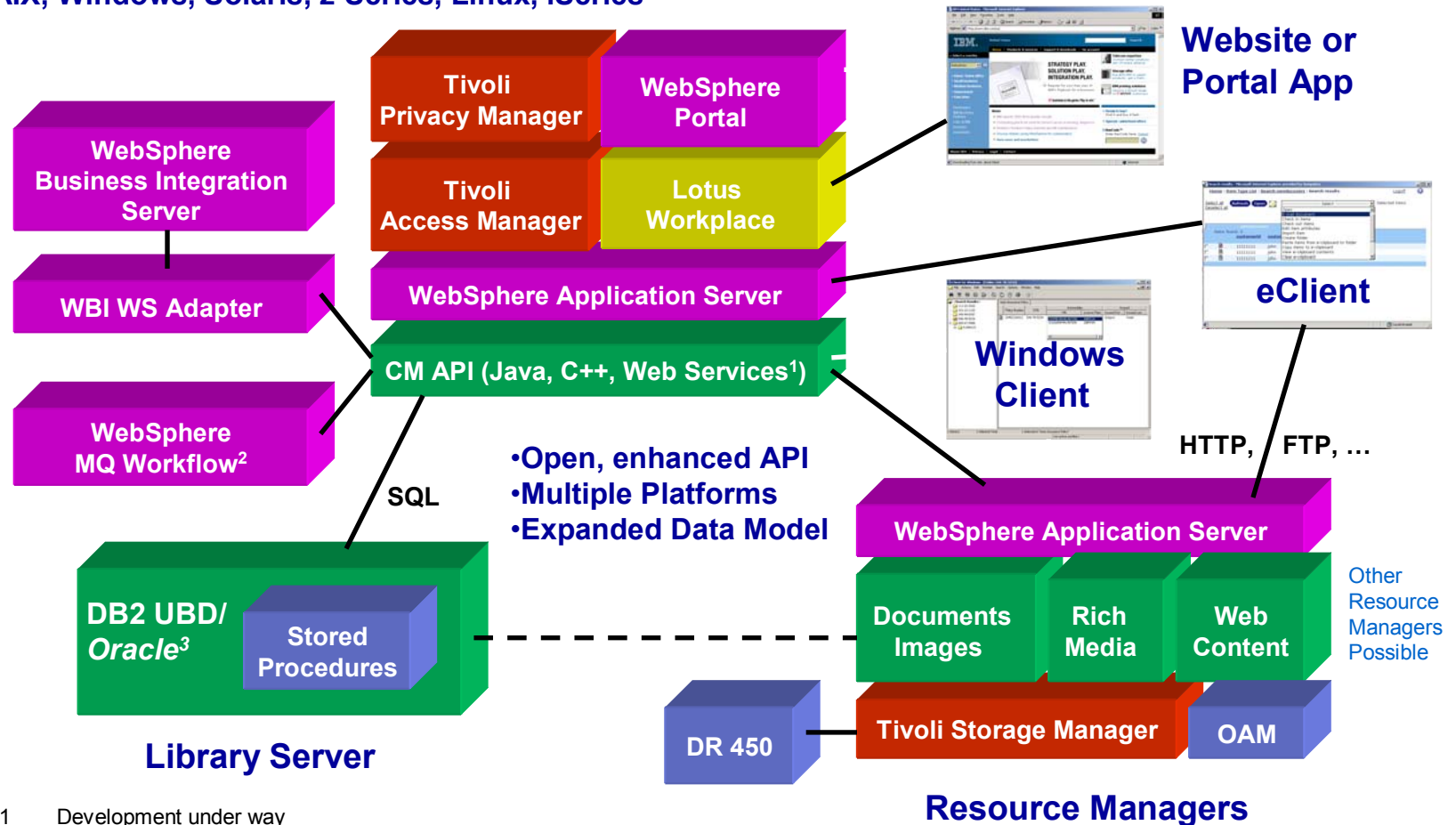
- ▲ Objectives
- ▲ Architecture review
- ▲ Data Modelling
- ▲ Features / Functions
- ▲ Migration

CM V8 Design Objectives

- ▶ Application-neutral, extensible, and scaleable
 - f* Beyond production imaging or document management
 - f* Enable Enterprise Content Management
- ▶ A physical data model describing the "building blocks"
 - f* Abstractions for modelling content, relationships
- ▶ Add high-level APIs as needed
 - f* To support different application models and standards
 - f* Direct mapping to the physical data model
- ▶ Push function down the stack, including into DB2, as required for performance, data integrity, or to leverage strength of other technologies
- ▶ Focus on performance
 - f* End user response time
 - f* Scalability

DB2 Content Manager Architecture

AIX, Windows, Solaris, z-Series, Linux, iSeries¹



- 1 Development under way
- 2 Packaged as II for Content Advanced Workflow Service
- 3 Currently under beta

CM Data Model Entities

▲ CM V8 Attribute

- ▶ Holds persistent state of runtime entities
- ▶ Attribute have properties: length, min / max value, search'ble....

▲ CM V8 ItemType

- f* Basic entity or schema – set of attributes and properties

▲ CM V8 Item

- f* An instance of an ItemType
- f* A record in a "component" table representing a document, folder, video, or other business object

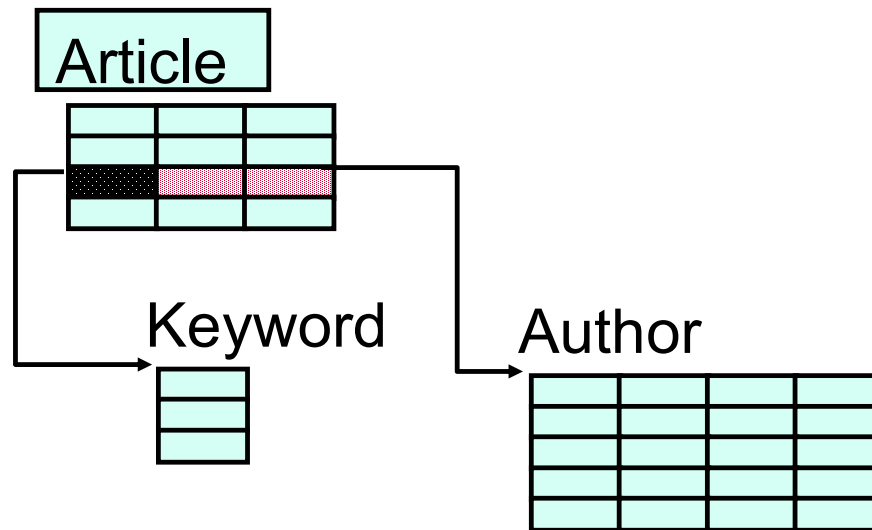
Data Model Example

*f*Entity: Article

An Article is represented by Attributes such as:
Title, Subject, Date, etc.

Article		
Title	Subject	Date

The state of entities is stored in
single – and multi-valued attributes

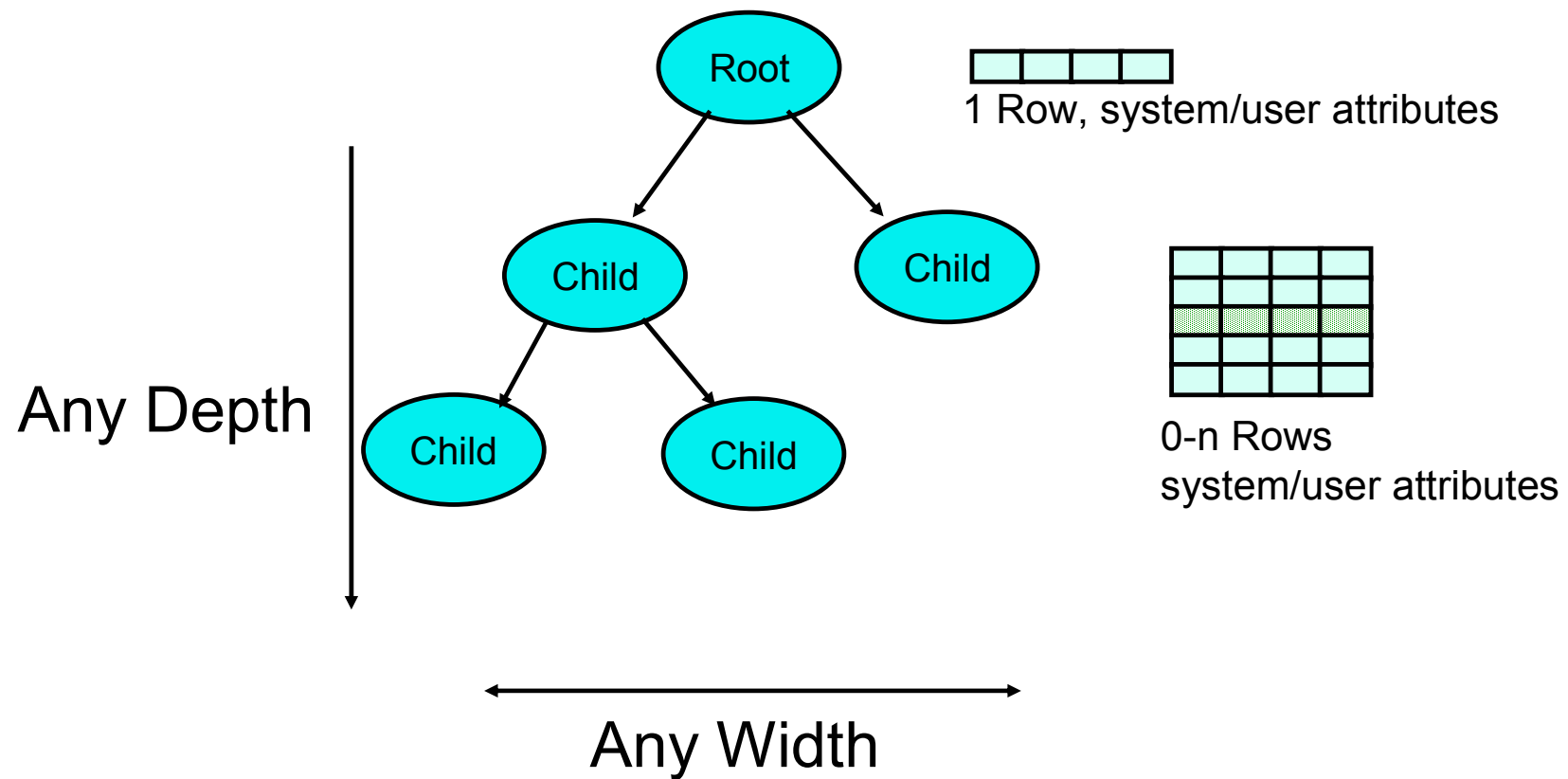


Articles have Authors

Last name
First name
Company
Address -- how many?

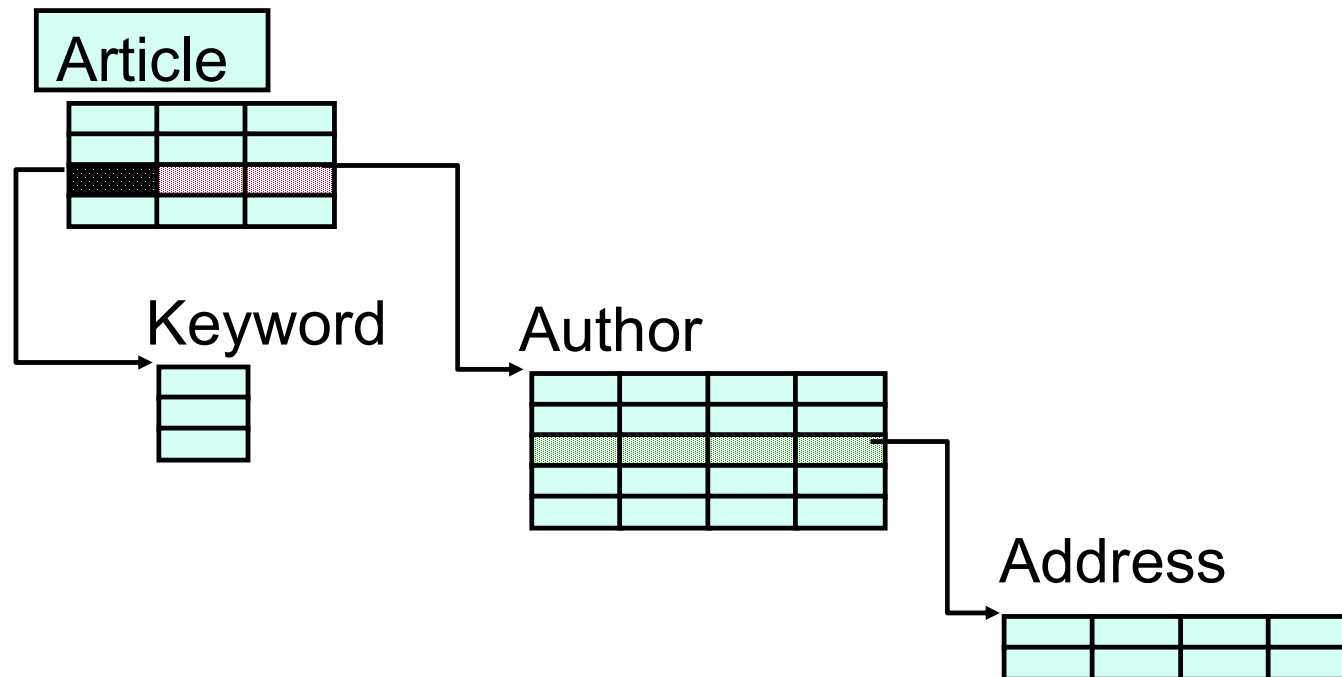
Articles have Keywords

Attributes are organized in a hierarchical structure: CM Item Type



CM Items: an instance of an entity

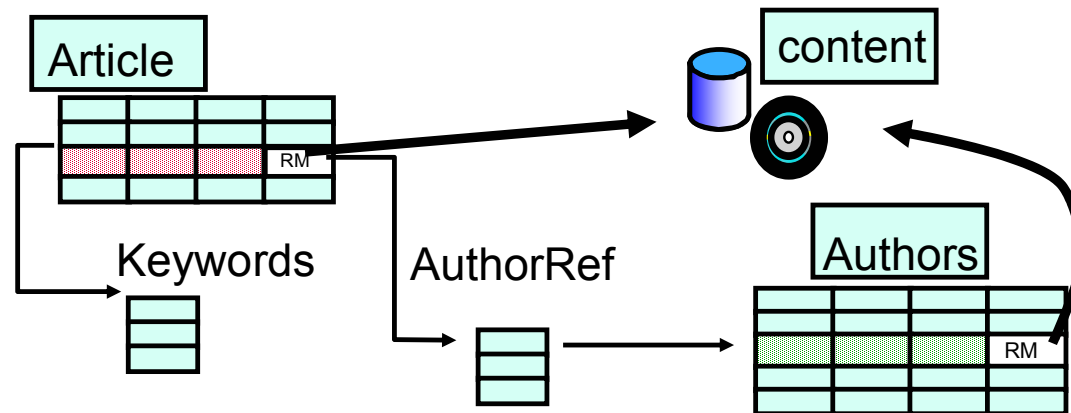
One row in possibly many tables



Authors have 1 or more addresses

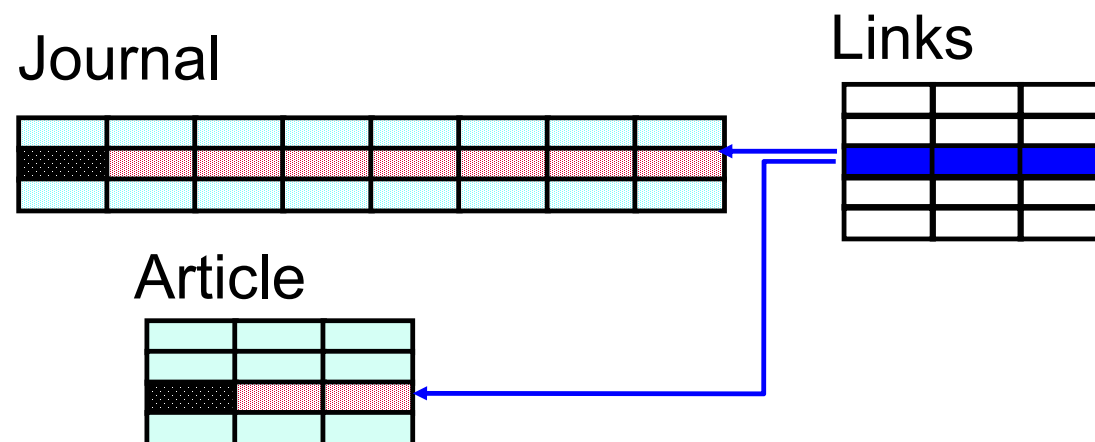
Modeling relationships among entities: Associations using Reference Attributes

- ▶ What if Authors have Photographs or one Author is associated with multiple Articles?
- ▶ A Reference attribute creates a relationship between records
 - f* Points to version of an item from root or child component
- ▶ Often a better choice than hierarchical ItemType
 - f* Eliminates duplication of data
- ▶ "Delete Restrict" ensures referential integrity



Modeling relationships among entities: Aggregation and composition using links

- ▶ Used to represent a relationship between two Items
 - f* Such as Document in Folder
- ▶ Items may be linked to/from 0-N Items, nested to any level
 - f* Cycles or loops allowed
- ▶ User defined Link Types supported for other semantics
 - f* Maps and Graphs in Info Mining



Modeling relationships among entities: Associations using foreign keys

▲ DB2 foreign keys for data validation

f If value being stored in CM table doesn't exist in target table,
an error is returned

f CM table to CM or customer table

Authors

UserID	Last Name	First Name	Title

ICMSTUSERS (CM User table)

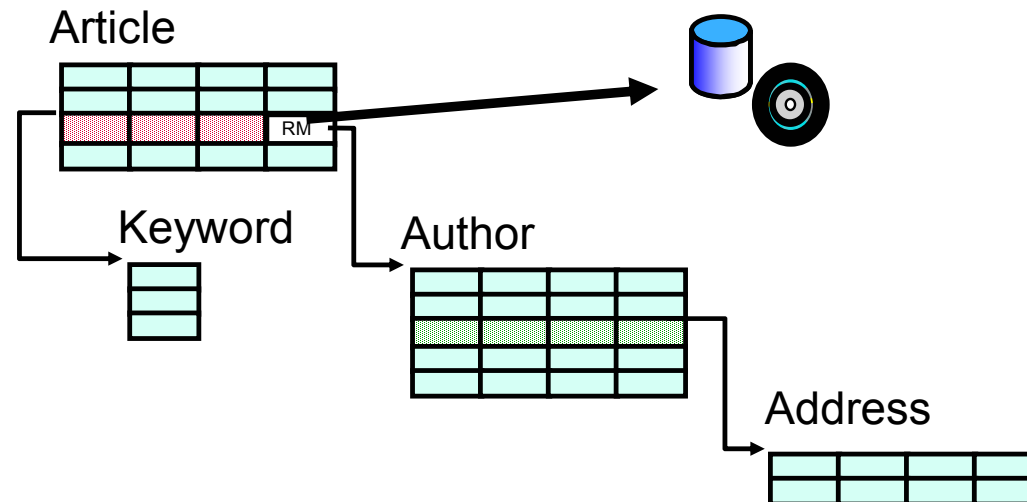
UserID	DomainID	PrivSet

Advantages and restrictions of linking mechanisms

Linking mechanism	Used at component level	Linked elements can be deleted	Limited by version
Link	Root to root	Yes	No
<i>Reference</i>	<i>Root or child to root</i>	<i>Specify when you create the reference</i>	<i>Specify when you create the reference</i>
Foreign key	<i>Root to a different item type or external table</i>	<i>Specify when you create the foreign key</i>	<i>Specify when you create the foreign key</i>

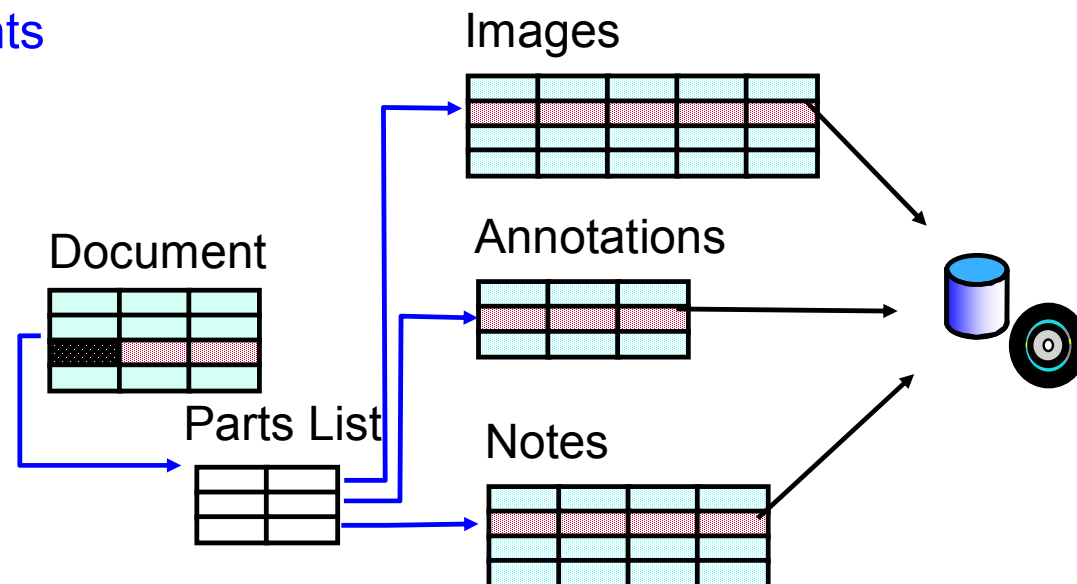
CM V8 built in entity types: Resource

- ▶ An Item with additional System Defined attributes defining the location, type, size, etc of an Object
 - f* User attributes, ACL, expiration date, etc
- ▶ Ideal for representing single-part documents
- ▶ Only a "root" component can point to an object



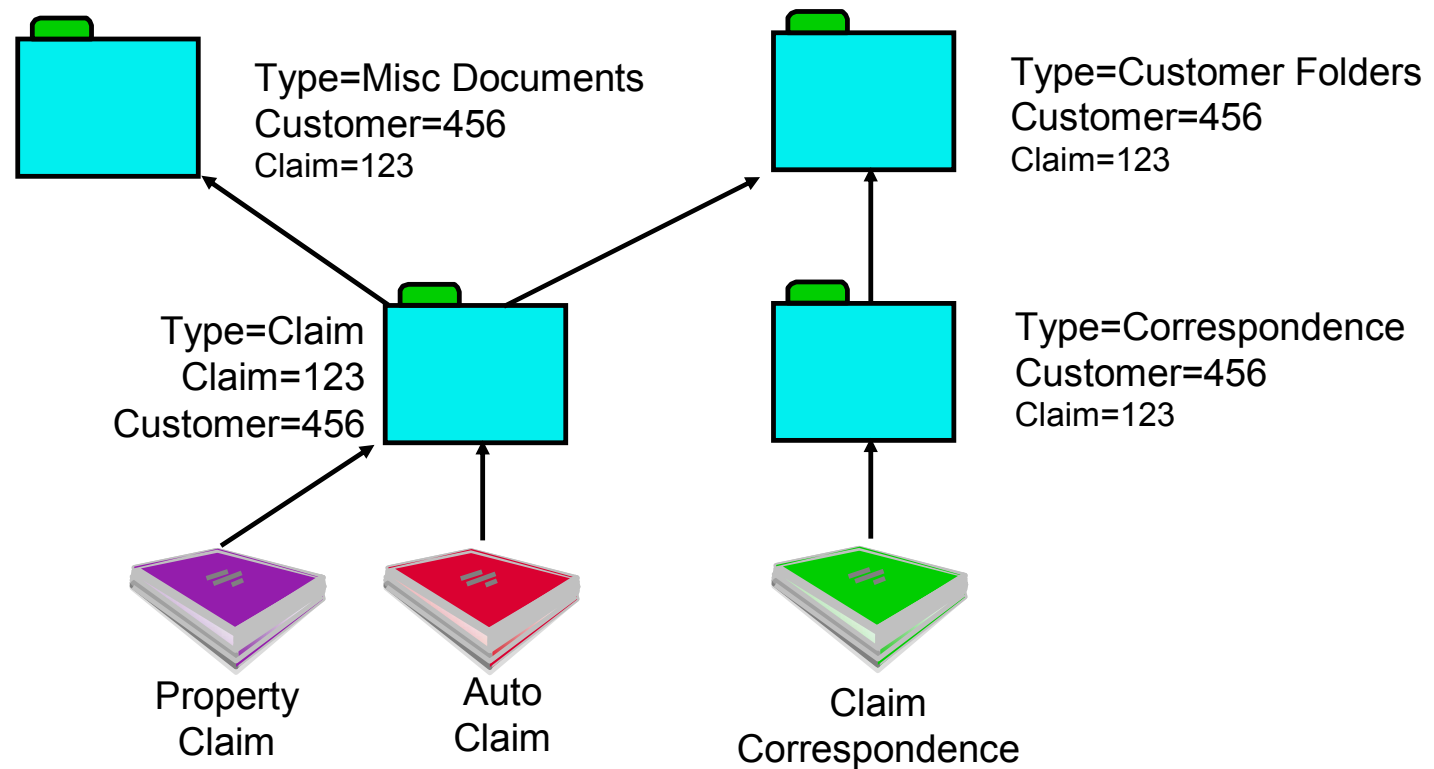
CM V8 built in entity types: Document

- ▲ Document Model -- using the building blocks
- ▲ Control of Part Types per Document Type
 - f* For example: Image and Notes, but no Annotations
- ▲ Versioning rules per Part Type
- ▲ ACL per Part Type or Part
- ▲ Enablement for advanced applications
 - f* Attributes on parts
 - f* Compound Documents



Auto Foldering

- ▶ When an Item is created, add it to a folder
- ▶ Implemented in the server vs. client
 - f* multi level and multi attribute



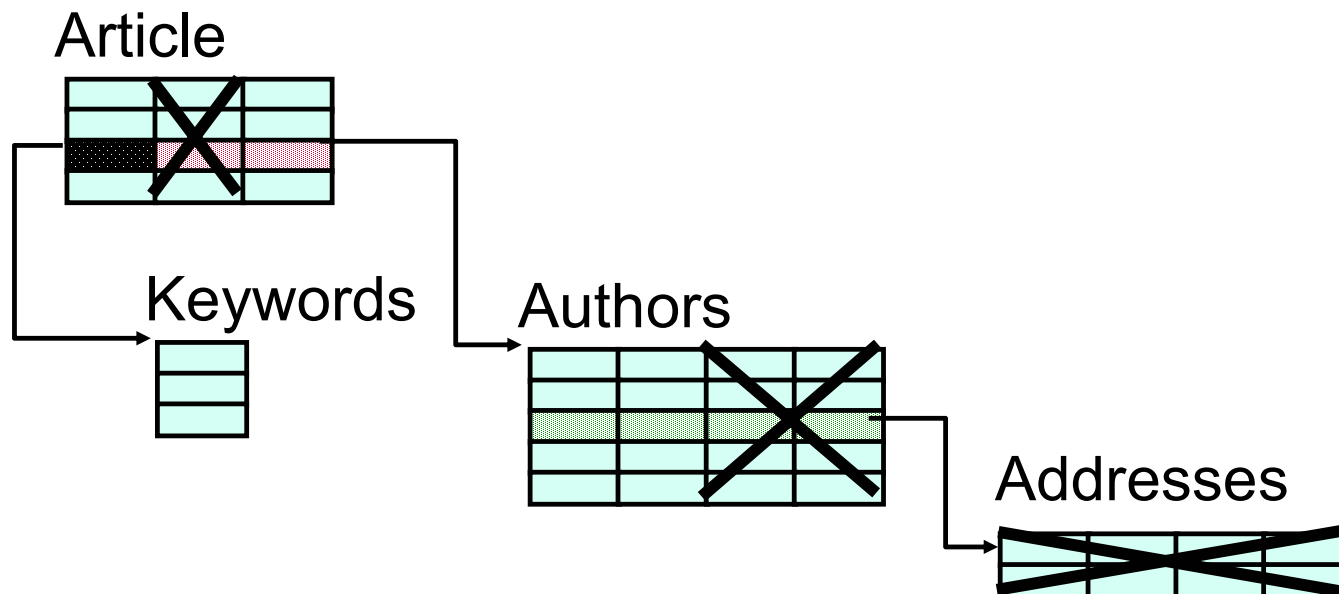
Views

▲ DB2 Views are created for each table to support read-only access with SQL

f Access control logic built into view definition

f Intended for use by Host or Workstation applications

f Can be joined with non ~~view~~ tables in queries



CM Access Control

- ▲ Access control is an integral part of Content Manager
- ▲ Access Control List (ACL)
 - f* One or more Users or Groups and a Privilege Set
 - f* Privilege Set is a list of privileges
 - Create, Retrieve, Update, Delete, etc. for Items
 - Add User, Modify ACL, etc. for SysAdmin operations
- ▲ Access control may be at the Item Type level
 - f* Each type of document could have a different ACL
- ▲ Or at the Part Type level
 - f* Scanned image, Annotation, Notes can have different ACL for each type of document
- ▲ At the Item level
 - f* Each document could have a different ACL
- ▲ Exits provided in to enable authorization through UDF call

Administrative Domains

- ▲ Users may be defined as "domain administrator"
 - f* Add, modify, delete users or manage groups within the domain
 - f* Only groups, RMs, Privilege Sets, etc. in the domain may be assigned by a domain administrator



System Administrator
Authority based on Privileges



Department Admin



Department Admin

Transactions

- ▲ BeginTransaction, EndTransaction
 - f* Uses DB2 for Library Server
 - f* Resource Manager is included the transaction scope
- ▲ Eliminate orphan or missing objects
- ▲ Ensure consistency between LS and RM
- ▲ "Explicit" transactions can include any number of actions
 - f* But locking of rows can impact performance or
- ▲ "Implicit" transactions ensure LS/RM integrity without work by the application writer

Versioning

- ▲ Some applications require that the original document be preserved and modifications result in new versions
- ▲ Administrator will specify per ItemType
 - f* Always, Never, Application control
 - f* Maximum number of versions to maintain
- ▲ Meta-data and parts (objects) can both be versioned
 - f* Document can be versioned without copying objects

Retention Management

- ▲ Each ItemType can have a retention period
 - f* Expiration date calculated when Item is created
 - f* Can be updated by applications
- ▲ Removal performed by custom application or client
 - f* Complete solution is provided by IBM Records Manager

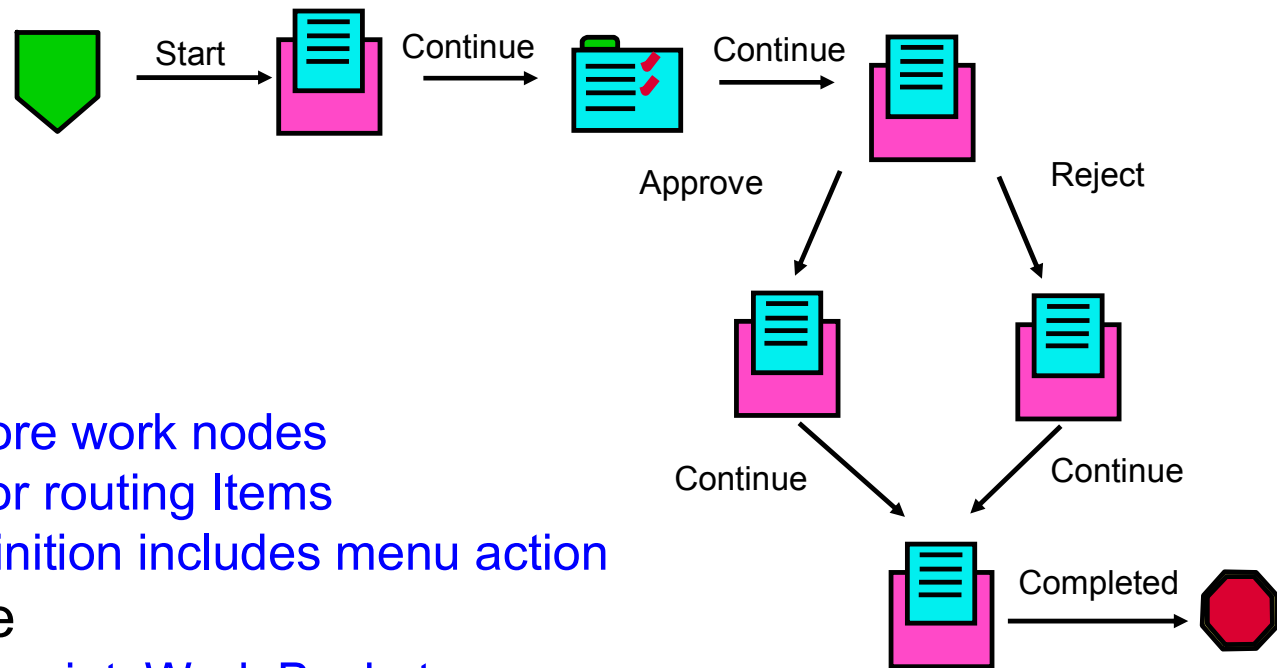
History/Event Logging

- ▶ Who did what to which CM entity and when ?
- ▶ Store history/events/audit records in a table
 - f* Enables queries using SQL
 - f* Workflow events will be in CM tables
- ▶ Server logging will be controlled by administrator
 - f* Per ItemType, records for Create, Retrieve, Update, Delete
- ▶ APIs provided for client applications to write to or query the Item event log
 - f* Reports will be generated via SQL queries
- ▶ Records are also written for events such as Resource Manager state change (available/unavailable)
- ▶ If configured, can include count of “concurrent” users

Document Workflow

- ▲ Small footprint, High performance
- ▲ Single definition of users, privileges
- ▲ ACL checking on routed Items
 - f* Work Lists contain only documents/folders to which user has access
 - f* Count of work is accurate based on ACL
- ▲ No "out of synch" conditions
- ▲ Document Routing operations can be included in a CM Transaction
- ▲ Single audit trail for Item create, retrieve, update, delete, and routing

Document Workflow



▲ Process

- f* One or more work nodes
- f* "Target" for routing Items
- f* Route definition includes menu action

▲ Work Node

- f* Collection point, Work Basket

▲ Work list

- f* "Source" for obtaining work

Document Workflow

▲ Collection point

- f* A Work Node at which a Folder waits for arrival of other Items
- f* 0 or more Items of 1 or more types

▲ Notification

- f* Work which has remained at a Work Node longer than expected
- f* Used for filtering in Work Lists

▲ Suspend for time and/or Items

- f* Manual or automatic resume
- f* Suspended work stays at the Work Node, is filtered by Work List

▲ Work List

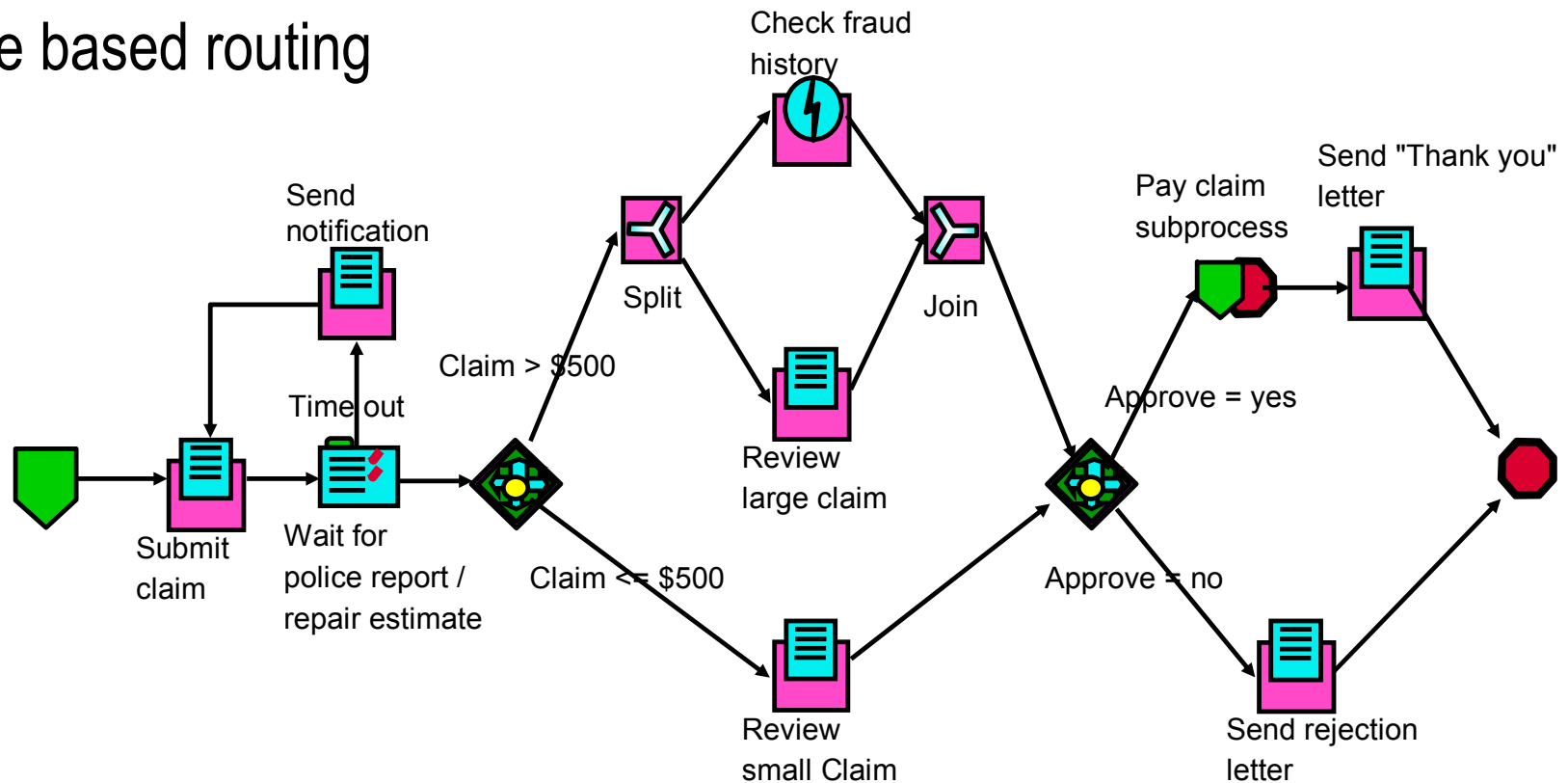
- f* 1 or more Work Nodes
 - A Work Node may be in multiple Work Lists
- f* Return up to "n" work packages with ordering, filtering
 - 1 work package = "system assigned"
 - Order by priority, date
 - Filtering -- Notify state, Suspend state, Assigned user

▲ Server exits on entering/leaving work nodes, Overload

Document Workflow new in V8.3

Event based routing

Value based routing



IBM Content Manager Version 8

- ▶ Rich function to meet the requirements of many types of applications
- ▶ Extensible architecture to meet future requirements
- ▶ Exploiting the strengths of strategic IBM technologies
- ▶ Achieving both performance and scalability

The foundation for
Enterprise Content Management